

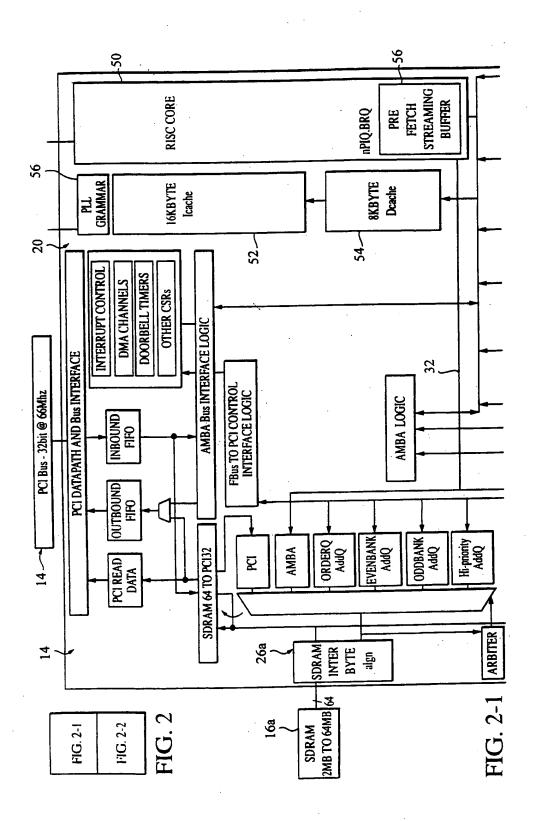
FIG. 1



Appln No.: 09/811,995 Page Applicant(s): Matthew J. Adiletta et al. REGISTER INSTRUCTIONS FOR A MULTITHREADED

Page 2 of 6

**PROCESSOR** 

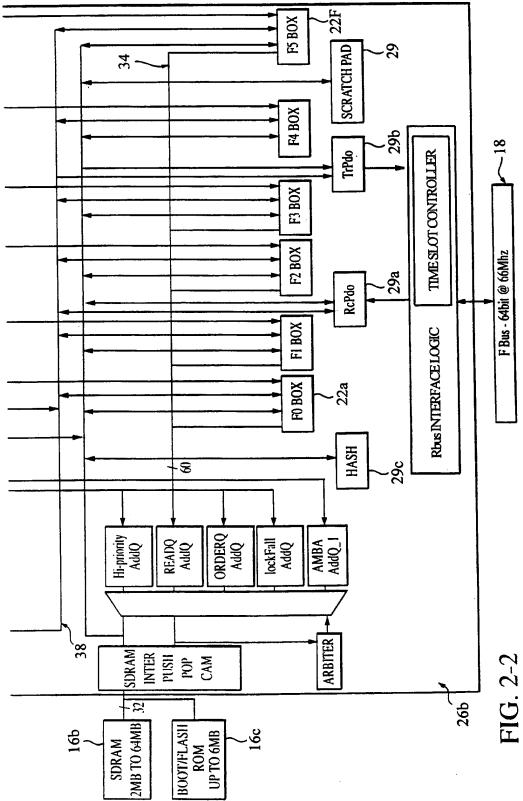


Appln No.: 09/811,995

Applicant(s): Matthew J. Adiletta et al.
REGISTER INSTRUCTIONS FOR A MULTITHREADED

Page 3 of 6

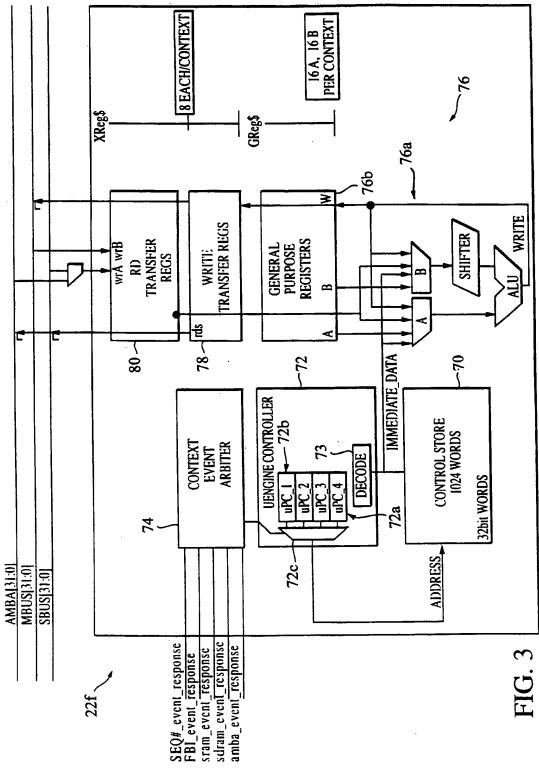
**PROCESSOR** 





Appln No.: 09/811,995 Page Applicant(s): Matthew J. Adiletta et al. REGISTER INSTRUCTIONS FOR A MULTITHREADED PROCESSOR

Page 4 of 6





Appln No.: 09/811,995 Page 5 of 6 Applicant(s): Matthew J. Adiletta et al. REGISTER INSTRUCTIONS FOR A MULTITHREADED

PROCESSOR

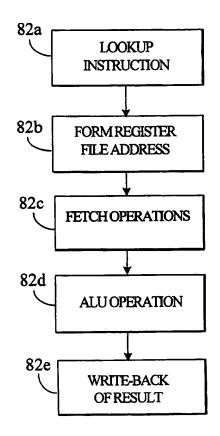


FIG. 4



Appln No.: 09/811,995

Applicant(s): Matthew J. Adiletta et al.

REGISTER INSTRUCTIONS FOR A MULTITHREADED

**PROCESSOR** 

ALUop ALUop **ALUop** 7 7 Srl im Bi 0 4 4 4 B S S 5 Ω 5 Ó 9 9 9 7 loB Abs Sec B rel source rel source 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8  $\infty$  $\infty$  $\infty$ immediate 6 20 19 18 17 16 15 14 13 12 11 10 9 9 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 8 A rel source A rel source A rel source A absolute source rs amonnt amount amount SΨ 22 21 31 30 29 28 27 26 25 24 23 O O sw shift rel dest reg 0 0 sw shift | rel dest reg rel dest reg dest reg O O sw shift 1 0 0 ALU/SHIFT (set cc) ALU/SHIFT (set cc) ALU/SHIFT (set cc) **ALU/SHIFT** (set cc)

Shift Decode:

(rs,r0) decode ([31:0] shifts into [63:32] and take [63;32]):

00 = left rotate 01 = right shift (32-ShfAmt = Right Shift Amt)

10 = left shift

11= double shift ( upper A-op shifts into lower B-op)

===> "left rotate" of zero gives zero shift (therwise zero amount signifies indirect shift)

ALU-OP decode:

1111= A+B+Cin 1101 = A + B(16)100 = A + B(8)1110 = A + B1000 = A - B1001 = B - A1010 = 1011= 0100 = ~A&B (~and) 0101 =XOR 0111= mul-stuff 0110 = OR0011= A&~B (and~ 0010 = A&B (and) 0001 = -B0.000 = B

Page 6 of 6